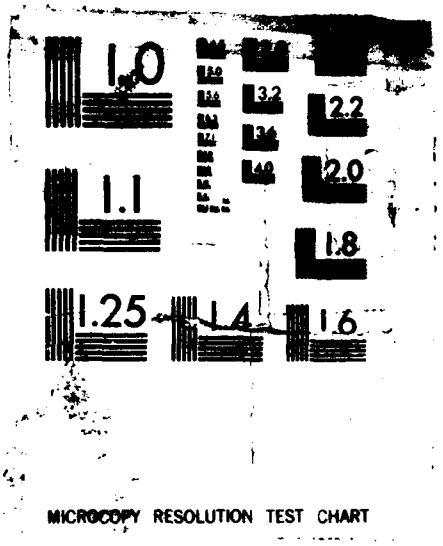
AD-A182 356 SHIPS HIGH-LEVEL PROCESS SIMULATION FOR VLSI (VERY LARGE SCALE INTEGRATED) MANUFACTURING(U) ELECTROCHEMICAL SOCIETY INC PENNINGTON NJ 1/1

UNCLASSIFIED S D LEEKE ET AL. 28 FEB 87 MDA903-80-C-0432 F/G 9/1 NL


1.0
1.1
1.25
1.4
1.6
2.2
2.0
1.8
MICROCOPY RESOLUTION TEST CHART

75-Word Abstract Form

Extended Abstract must be submitted with the 75-Word Abstract by Feb. 28, 1987

Honolulu, Hawaii—October 18-23, 1987

Submit to: The Electrochemical Society, Inc.
10 South Main Street, Pennington, NJ 08534-2896
With a copy to the Organizing Chairman

Abstract No. .....
(to be assigned by the Society)

Schedule for ..... Automated IC Manufacturing (AIM-III)
(Title of Symposium)

Sponsored by ..... Electronics .....
(Division/Group)

Title of paper SHIPS: High-level Process Simulation for VLSI Manufacturi.....

.....

Authors (Underline name of author presenting paper.) Steven D. Leeke and Dr. Krishna Saraswat

.....

Business affiliation and address The Center for Integrated Systems

Stanford University

Stanford, CA 94305 (415) 725-3600 x96
(State or Country) (ZIP Code) (Telephone No.)

VERY LARGE SCALE INTEGRATION

~~VLSI~~ fabrication is a complex set of operations set within a rapidly changing technology environment. It is subject to chance variations in process parameters, poor understanding and control of many parameters, and subtle interactions between them. At the same time, processing times and costs are increasing. The natural side-effect of these is that IC manufacturers must require more evidence that the predicted and actual process characteristics will be within a tight tolerance. The prevailing means to verify process performance is through simulation. Unfortunately, while traditional process simulation tools such as SUPREM III have become pervasive in the IC industry as design aides for process development, they are not suited to the problems of manufacturing-level process simulation. What is required is a high-level process simulation tool that can introduce abstractions that reduce complexity and algorithms that reduce simulation costs. We have built such a tool called SHIPS, which stands for Stanford High-Level Incremental Process Simulator. The SHIPS tool provides a high-level process description language, a structured user-interface, and a new simulation algorithm for reducing the number of modules that must be simulated in analyzing the manufacturability of a process. Our implementation uses the SUPREM III process simulation program while incorporating a simulation engine independent architecture. In this paper we describe the SHIPS system and the algorithm it uses to achieve large reductions (69% in the example given) in the amount of CPU time required to perform variational process simulation for analyzing the manufacturability of a process.

Do you require any audiovisual equipment?

- [X] 35 mm (2 x 2 in.) slide projector
- [ ] Overhead projector
- [ ] Specify other (subject to availability and cost)

Has the information in this abstract been presented verbally, submitted for publication, or published?

[ ] Yes [X] No

If the answer is yes, please provide the reference (except in the case of invited review presentations). _____

Is a full length paper on this work to be submitted for Society Journal publication? [X] Yes [ ] No

Papers presented before a Society technical meeting become the property of the Society and may not be published elsewhere without written permission of the Society. Papers presented at Society technical meetings must be authored by a member or sponsored by an active member.

Dr. Krishna Saraswat
Insert name of Society member author or sponsor

SHIPS: High-level Process Simulation for VLSI Manufacturing

Steven D. Leeke and Krishna Saraswat
The Center for Integrated Systems
Stanford University
Stanford, CA 94305

1. Introduction

VLSI fabrication is a complex set of operations set within a rapidly changing technology environment. It is subject to chance variations in process parameters, poor understanding and control of many parameters, and subtle interactions between them. At the same time, processing times and costs are increasing. The natural side-effect of these is that IC manufacturers must require more evidence that the predicted and actual process characteristics will be within a tight tolerance. The prevailing means to verify process performance is through simulation. Unfortunately, while traditional process simulation tools such as SUPREM III have become pervasive in the IC industry as design aides for process development, they are not suited to the problems of manufacturing-level process simulation. They are designed for single-run simulation, not variational, and have no support for gathering statistics on multiple simulations. What is required is a high-level process simulation tool that can introduce abstractions that reduce complexity and algorithms that reduce simulation costs. We have built such a tool called SHIPS, which stands for Stanford High-Level Incremental Process Simulator. In this paper we describe the SHIPS system and the algorithm it uses to achieve large reductions in the amount of CPU time required to perform variational process simulation for analyzing the manufacturability of a process.

2. High-level Process Simulation

To date there have been only a few efforts at manufacturing-level process simulation. That is, the application of process simulation to problems of variability in a process. This requires the statistical analysis of the process via a large number of simulations. Some of the better known and successful efforts have come from CMU and Hitachi, with the Fabrics family of programs and the CASTAM program, respectively[1, 2, 3]. The former is an evolving set of tools for process information capture and simulation, device simulation, and parameter extraction for circuit simulation. The Fabrics process simulation is done using only analytic models. This has the advantage of speed, but the disadvantage of significant errors for state-of-the-art technologies.

3. The SHIPS System

The SHIPS system improves on previous work by:

- using the SUPREM III process simulation program and its advanced physical models.[4]
- utilizing an incremental simulation algorithm to reduce the overall CPU time required for simulation.
- utilizing a high-level process simulation language, SHIPS, that provides a very compact notation for representing complex process simulations.
- including a three level abstraction of VLSI processing to reduce the complexity at any one level. The three levels are, in descending order, *process*, *module*, and *step*.
- using process module libraries to augment the processing abstraction. This provides a way to group related collections of processing modules. This is similar to the use of standard cell libraries for ASICs.
- providing a structured user interface that is menu-driven, uses error-checked form-based entry, and includes extensive on-line help. This reduces the level of expertise required to profitably utilize the system.
- performing the simulations in a simulation *engine* independent manner. This amounts to having the SHIPS program generate code for the process simulator and handling all interfacing to the simulator itself. This allows the user to concern himself with the specification of the process and the simulation and its results, rather than with the intricacies of the simulator. It also supports the use of more than one process simulator for a given simulation.

4. The SHIPS Language and Compiler

The SHIPS language is a structured language designed to provide a compact notation for representing large-volume process simulations. The language has constructs for defining the process to use, the views of the process to be simulated, and the output information of interest. A view is, in the simplest sense, a certain process sequence as determined by the lithography of the process. For a one-dimensional simulator such as SUPREM III the natural view is a profile, while for a two-dimensional simulator it becomes a cross-section. (See figure 1.)

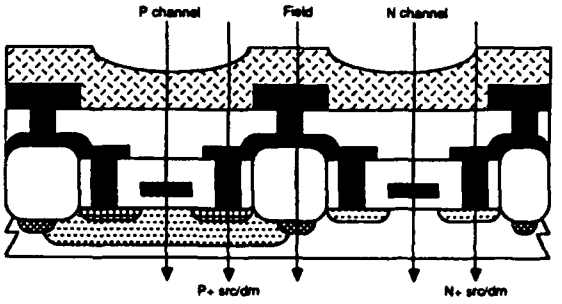


Figure 1: The views of a process using a 1-dimensional simulator.

The language supports numeric expressions, functions, and variables; string variables; a block structure based on the modules of a process; for-to and set assignment statements for multi-value assignments; simple statistical and factorial analysis of the simulation results; and integration to other simulation programs for post-processing of the simulation results via output statements and yield parameters. An example of a simulation written in the SHIPS language is shown in figure 2.

```
SIMULATION Raml_Process; /* The name of the simulation */
PROCESS 2u_cmos(p_channel,n_channel,p_src_drn,n_src_drn);

VAR /* Variable declarations */
     iox_time : NUMERIC;
     iox_step : NUMERIC;
END;

iox_time = 100;
iox_step = 10;

/* WITH statements operate on a module to change the
   parameters within the module. */

WITH Initial_Oxidation DO
     /* A multiple assignment using a FOR statement */
     FOR time1 = iox_time TO 140 ADD iox_step;
END;

WITH P+_src_drn_implant DO
     /* A simple assignment */
     energy = 100;
     /* A multiple assignment using a SET statement */
     SET dose = [1e13,1e14,5e14];
END;

OUTPUT jctn_depth mean stddev; /* The results we want */
SIMULATE FROM init TO passivate; /* Simulate */
```

Figure 2: An example simulation program in the SHIPS language.

5. The Incremental Simulation Algorithm

Simulating the effects of variations in process parameters can become an imposing task when either the complexity of the process is high, the number of parameters being varied is large, or the number of values assigned to any one parameter is large. At present, most full process simulations using a physical simulator such as SUPREM III are done in an a priori fashion. That is, the *entire* simulation is repeated each time a parameter is changed. Incremental simulation utilizes the modularity of the process description to reduce the simulation to the minimum number of modules possible. The simulations are generated such that only those modules which will be impacted by a particular parameter change are re-simulated. Keeping track of these intermediate simulations requires a simulation supervisor.

The incremental simulation algorithm provides a dramatic improvement over a priori simulation. The following definitions are used in expressions comparing the number of modules simulated during a priori and incremental simulation.

- M = The length of the process in modules
- I = The number of modules that were changed such that the parameters of that module have a cumulative product of changes that is greater than one. Using the definitions below, this is true of the kth module in the process if $\prod_{p=1,P_k} V_{pk} > 1$.
- P_i = The number of parameters assigned two or more values in the ith module that was changed
- V_{ip} = The number of variations of the pth parameter in the ith module that was changed

- δm_i = The number of modules simulated in the ith level of the incremental simulation
- S_{old} = The number of modules simulated by creating all of the simulation input programs a priori
- S_{new} = The number of modules simulated by doing the simulations incrementally

Given these definitions the following expressions can be written for the total number of modules simulated with the a priori and incremental algorithms.

- $S_{old} = M\{\Pi_{i=1,I}(\Pi_{p=1,P_i} V_{pi})\} = \Sigma_{i=1,I}[(\Pi_{j=1,I}(\Pi_{p=1,P_j} V_{pj}))\,\delta m_i]$
- $S_{new} = \Sigma_{i=1,I}[(\Pi_{j=1,I}(\Pi_{p=1,P_j} V_{pj}))\,\delta m_i]$

Making the following definitions:

- $A_i = \Pi_{j=1,I}(\Pi_{p=1,P_j} V_{pj})$
- $B_i = \Pi_{j=i+1,I}(\Pi_{p=1,P_j} V_{pj})$

we find that $S_{old} / S_{new} = \Sigma_{i=1,I} A_i B_i \delta m_i / \Sigma_{i=1,I} A_i \delta m_i$. This implies that $\max(B_i) \geq S_{old}/S_{new} \geq \min(B_i)$ and in general $S_{new} < S_{old}$ for $I > 1$ and $S_{old} = S_{new}$ for $I = 1$. The gains achieved using the new algorithm, however, are dependent on where the variations occur in the process and their number.

From the algebraic results it can been seen that the reduction in simulation effort increases as the variations occur later in the process and in greater number. This is a result of the incremental nature of the algorithm. This can be seen more clearly in graphical form. In figure 3 the the differences between the a priori simulation algorithm and the incremental are highlighted. This figure uses the data given in the example below. Nodes represent simulations and connecting lines indicate that the node higher in the tree supplies the starting information for the simulation in each of its children nodes. The information pertaining to the a priori algorithm is shown in grey and to the incremental algorithm in black.

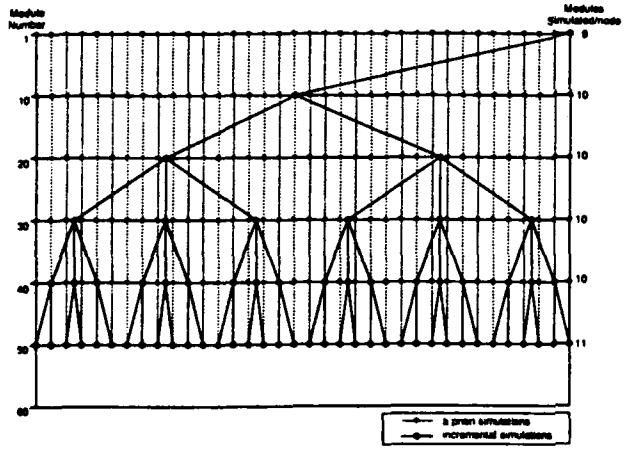


Figure 3: A comparison of the a priori and incremental simulation costs.

Only one-half of the simulations are shown since for the incremental algorithm the resulting tree of simulation nodes is identical about the root node. The number of modules simulated for each node is given at each level in the diagram along the right side. There are 216 grey nodes shown for a total of 432, indicating the simulation of 4320 modules. There are 64 black nodes shown (63 non-root) and when the numbers of simulations are added up for each black node, including those nodes not shown, the total is 1341 modules. This graphical representation is an easy way to see the advantages of the incremental simulation algorithm over the a priori.

In addition to reducing the cost of the full simulations, the SHIPS system is being extended to exploit the incremental algorithm for run-time control over the simulations. This will allow the user to specify limits for the values of interest in the simulations, and if those limits are exceeded for any given node the simulations represented by its descendant nodes will not be performed. This can be used to further reduce the cost of variational process simulation by reducing the size of the solution space.

6. Incremental Simulation Example

For a process with 60 modules, table 1 shows the modules that have been changed and the number of changes.

Module No.	Parameters Changed	No. of Changes
10	1	2
20	1	2
30	1	3
40	1	3
50	1	2

Table 6-1: Incremental simulation algorithm example for M = 60.

The results for the simulations are as follows:

- S_{old} = 4320 modules simulated
- S_{new} = 1341 modules simulated
- $S_{old} / S_{new} = 3.22$ and $S_{new} / S_{old} = 0.31$

This shows that new incremental simulation algorithm requires only 31% as much simulation effort as the a priori algorithm.

7. Conclusion

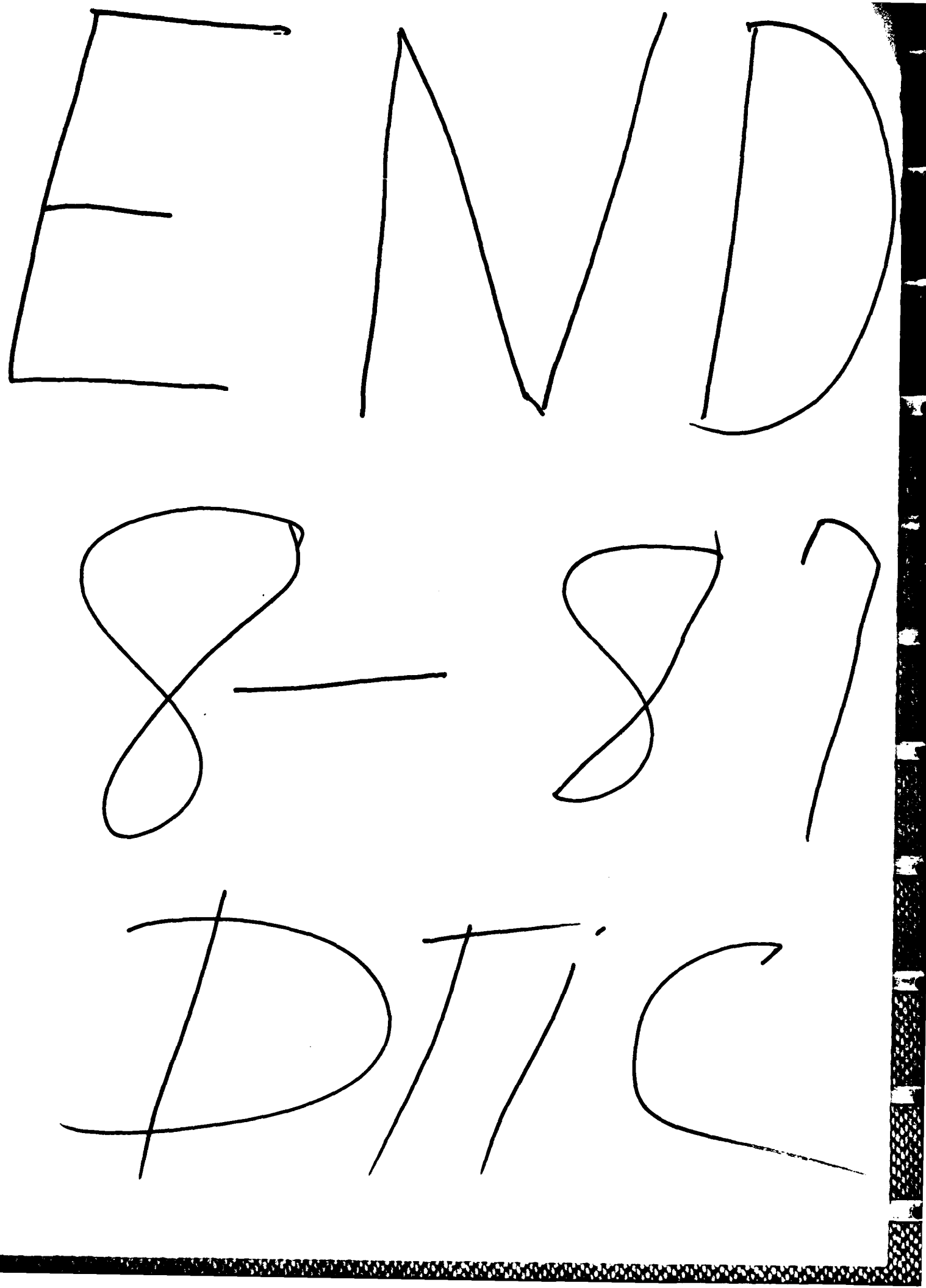
The SHIPS system for high-level process simulation provides a high-level process description language, a structured user-interface, and a new simulation algorithm for reducing the number of modules that must be simulated in analyzing the manufacturability of a process. Our implementation uses the SUPREM III process simulation program while incorporating a simulation engine independent architecture.

8. Acknowledgements

The authors wish to thank Dr. John Shott of the Stanford IC Lab for his leadership of the manufacturing simulation program. This work was supported by SRC contract 84-01-046 and DARPA contract MDA 903-80-C-0432.

References

1. Yukio Aoki, Tora Toyabe, Shojiro Asai, and Takashi Hagiwara, "CASTAM: A Process Variation Analysis Simulator for MOS LSI's", *IEEE Transactions on Electron Devices.*, Vol. ED-31, No. 10, October, 1984, pp. 1462-1467.
2. Sani R. Nassif, Andrzej J. Strojwas, and Stephen W. Director, "FABRICS II: A Statistically Based IC Fabrication Process Simulator", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems.*, Vol. CAD-3, No. 1, January, 1984, pp. 40-46.
3. Wojciech Maly and Andrzej J. Strojwas, "Statistical Simulation of the IC Manufacturing Process", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems.*, Vol. CAD-1, No. 3, July, 1982, pp. 120-131.
4. C.P. Ho, J.D. Plummer, S.E. Hansen, and R.W. Dutton, "VLSI Process Modeling - Suprem III", *IEEE Transactions on Electron Devices.*, Vol. ED-30, No. 11, November, 1983, pp. 1438-1453.


END
8-87
DTIC